Hi Andrew,

Contemplating the 1850 BUG and fidling with imaginairy laser-toolpaths I realised a flaw in the mindset discussing inside/outside toolpaths.

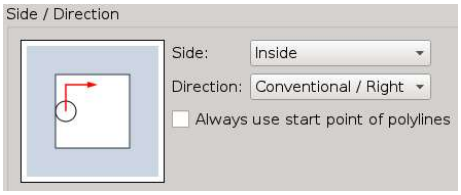
```
FS#1850 - Toolpath error
User who did this - Andrew (andrew)

----------
There are some issues here:
- Radius correction (inside / outside) is used for line fonts.
Inside / outside only makes sense for closed profiles.
- The line fonts that come with QCAD were originally made for
screen display and are not precise enough for effective CAM
processing. This will be adjusted in a future version, possibly
the next version.
----------
```

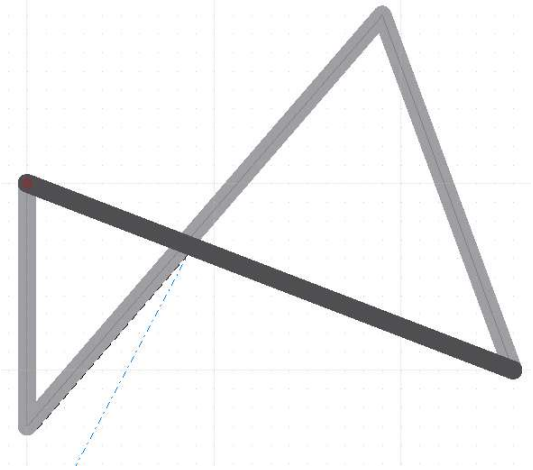
There are no inside/outside toolpaths. A toolpath is a line ***on*** a CAD-vector, or at an ***offset*** distance to the CAD-vector. Only in a cartesian flatland system, a rightside (positive offset) or leftside (negative offset) can exists, where the vector-direction is a positive value and the opposite direction a negative value.

In a 3D (real world) coordsystem an offset-distance can not be negative, so the 'parallel side' is then defined by a rotation around the vector; flatland-parameters might be used, as in rightside is zero degrees and leftside as 180 degrees.

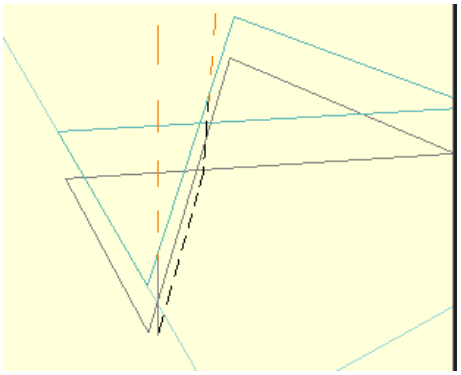
Concluding: Any line (in CAD or real world) can be offsetted at any side by a parallel line. Implying every single line (in 2D CAD too) should be parallel offsetted as **left** or **right**. This includes creating offset tool paths too.



Slot milling, as pocket milling, is **left** or / and **right** milling. Inside or outside are merely accidental humanlike definitions. As in the figure to the right defined by four coordinates, is there an inside and outside?



This flaw in mindset is very remarkable in the difference whenever selecting the On, Inside or Outside toolpath.



A nightmare if one tries to engrave these four lines at an offset distance.

The toolpath-errors I noticed on the ***line-fonts***, are created by the mindset of an existence of inside/outside toolpaths (and its assumption to the design of the G-code algorithm).

The whole purpose for creating line fonts, is the simple HPGL-translation for output to a penplotter. Creating line-fonts for screen displaying only seems a little bit unlikely. However, I can image the font-designer did not realise the implications of bad shape-definitions in the glyphs for generating G-code; testing the designed line-fonts on a penplotter or printer did look fine.